

GLOBAL POSITIONING SYSTEM HARDWARE KEY FOR SOFTWARE LICENSES

FIELD OF THE INVENTION

5 The present invention relates generally to software and/or hardware, and more particularly, to a license validation system which enables or disables software and/or hardware.

BACKGROUND OF THE INVENTION

10 Software piracy costs software manufacturers hundreds of millions annually in lost sales. Software piracy can take many forms. Although the most common form is to make unlawful copies of software, other forms include unlawfully enabling and using unpaid for software features in otherwise validly licensed software and using validly licensed software outside of permissible geographic parameters. In the latter situation, globally sold software
15 has different pricing structures based on different geographic regions. The differing pricing structures depend on a variety of factors, including exchange rate, demand, and the socioeconomic status of the region. For example, software are often sold at higher prices in the U.S., Japan, South Korea, and Europe but at lower prices in lesser developed countries such as China. Obviously, there is a great financial incentive to buy "gray market" software
20 in China when they are destined for use in the U.S., Japan, South Korea, or Europe.

 One method uses serial number information (e.g., medium access control or MAC address) to associate licensed software with corresponding hardware and uses differing hardware identifiers for differing price regions. In this method, a valid license file is required

to run a computational component. The license file contains a serial number that must be present on the hardware that is to execute the licensed software for the license to be valid and the software to be executable. In telecommunication applications, for example, the serial number of the control processor must be in the license file for the control processor to run the licensed software. The hardware identifiers are differing versions or ranges of the serial numbers, such that a specified region has a serial number of a particular format or within a particular range. When the system determines whether a valid license file is present, one of the checks is to determine whether or not the license has expired based on the system clock value and another is to determine for a stored region code set by the manufacturer that the serial number is correct for the corresponding region. This method is discussed in copending U.S. Patent Applications entitled "Securing Feature Activation in a Telecommunication System", Serial No. 09/357,679, filed July 20, 1999, to Serkowski; "License Modes in Call Processing", Serial No. 10/232,508, filed August 30, 2002; "Remote Feature Activator Feature Extraction", Serial No. 10/232,906, filed August 30, 2002; "Flexible License File Feature Controls", Serial No. 10/231,999; "License File Serial Number Tracking", 10/232,507; "Licensing Duplicated Systems", Serial No. 10/231,957; "Software Licensing for Spare Processors", Serial No. 10/232,647; "Temporary Password Login", Serial No. 10/387,182; and "Ironclad Notification of License Errors", Serial No. 10/405,176; each of which is incorporated herein by this reference, and is currently being implemented commercially in Communication Manager™ by Avaya, Inc.™.

The use of the system clock setting to determine whether or not the license has expired can be circumvented by the licensee. As will be appreciated, software licenses can be limited or unlimited in duration. Licenses that are limited in duration are often much less

Moreover, the use of a region code or location feature coupled with a specific serial number provides only weak security against geographic software piracy. The mechanism addresses only gray market hardware issues. It does not address gray market software concerns.

10

15 In one embodiment of the present invention, a method for validating an intended use
of a computational component is provided that includes the steps of:

20

(i) comparing the geographic location with the predetermined geographic location permitted by the license; and

(ii) comparing the clock setting with the expiration date of the license; and

(c) when the geographic location is not a permitted geographic location under the license and/or when the clock setting is outside of the permissible term of the license, determining that use of the computational component is not permitted. When the clock setting is within the permissible license term, the geographic location is a permitted geographic location under the license, and/or other criteria are satisfied, the intended use of the computational component is valid. By using the GPS timing information to determine whether or not the license has expired rather than the system clock setting, the licensee is unable to extend the license beyond its otherwise permissible term. By determining the geographic location of the computational component and/or key device, the methodology can prevent the gray market use of the computational component.

The geographic location may be determined by any suitable technique, with a location determination by a Global Positioning System or GPS receiver being preferred due to its accuracy (within 100 meters), low cost, and ready availability. For more efficient processing, the GPS coordinates, which are typically expressed as a pairing of latitude and longitude, are converted into a region code by mapping the coordinates against a conversion table. The predetermined geographic location permitted by the license can be expressed as one or a number of region codes. Alternatively, the determination of proper geographic use of the computational component can be based on the absence of the region code corresponding to the current geographic location from a set of region codes. Typically, the GPS module is located in the key device, which is commonly configured as a dongle. The preferred type of dongle is a smart card.

The key device is typically required to activate the computational component. In

other words, the computational component cannot be operated or executed unless the key device is in communication with the component.

To ensure that a valid key device is used to activate or execute the computational component, the key device is preferably authenticated by the computational component using suitable authentication information (e.g., a serial number associated with the key device and/or computational component). To be successfully authenticated, the serial number provided by the key device must match a serial number stored in the computational component, typically in a license file. Unless properly authenticated, the key device is not recognized by the computational component.

The key device can be configured to prohibit operation over a large network. A dishonest user can set up a computational system such that the computational component is running in a different region or country than the region or country where the key device is located. Although having a key device in one region to activate hardware/software in another distant region is a great inconvenience, such license abuse is not inconceivable if the cost savings to the user are large.

To further illustrate this embodiment, an operational description of a preferred configuration of the embodiment is provided. In the configuration, the licensed application makes periodic smart card queries to validate the license. In response to a query, the smart card translates the location determined by the GPS receiver into a region code and returns the region code to the licensed application along with the serial number. In this configuration, the license file would specify the region code(s) in which the software is licensed to operate based on what the customer ordered. If the region code returned to the application were not an allowed region code as defined in the license file, the software would

not function. The smart card query from the licensed application can include the region code from the license. The smart card would then only return a serial number response if the GPS-determined position matched the region code in the query. Although such smart cards may not be inexpensive (approximately \$200 per copy by today's pricing), for medium and high value software the expense can be justified if the cost differential between regions is significant. GPS chips are expected to drop dramatically in cost as they become mass-market items in cell phones and other mobile devices.

These and other advantages and features of the invention will become apparent from the following description of the invention taken together with the drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram representation of a dongle according to an embodiment of the present invention;

Fig. 2 is a flow chart illustration of the operational steps performed by a license validation agent in a computational component seeking license validation;

Fig. 3 is a flow chart illustration of the operational steps performed by a validation agent in the dongle of Fig. 1; and

Fig. 4 is a block diagram representation of the dongle of Fig. 1 in communication with the computational component performing license validation.

DETAILED DESCRIPTION

Referring to Fig. 1, a dongle 100 of one embodiment of the present invention is depicted. The dongle 100 includes a port 104, a Global Positioning System or GPS module

108 coupled to an antenna 112, a processor 116, a memory 120 and an optional power source 124. As can be seen from Fig. 4, the dongle 100 communicates with a computational component 400 attempting to validate the existence of a valid license.

5 The port 104 of the dongle 100 typically connects to a port on the computational component 400 attempting to successfully validate the existence of a valid license for the intended operation of the computational component. The computational component 400 can be any entity capable of performing a task or executing instructions, *e.g.*, a logic-containing board or chip such as an application specific integrated circuit or ASIC, a (control) processor, software, *etc.* In one configuration, the computational component 400 is a license-controlled
10 telecommunication application. The port 104 may be adapted to be connected to a parallel port, a serial port, or any other type of port which may be available for data transfer, including a universal serial bus (USB) port.

The GPS module 108 can be any suitable software and/or hardware for receiving GPS signals and determining the current GPS timing information (*e.g.*, a clock setting including
15 one or more of time of day, day of month, month of year, and year) and the module's current location expressed in GPS coordinates (*e.g.*, a latitude and longitude pairing). The GPS module 108 is in communication with the antenna 112 to receive the GPS signals.

The processor 116 can be any functional unit that interprets and executes instructions or processes coded instructions and performs a task. A processor typically includes an
20 instruction control unit and an arithmetic and logic unit. Typically, the processor 116 is a microprocessor.

The memory 120 can be any suitable medium containing addressable storage space. The memory can be volatile storage and/or nonvolatile storage and can be read-only and/or

random access. Typically, the memory is in the form of EEPROM.

The power source 124, which is optional, can be any suitable power supply. The power source is optional in that power could be received entirely from the computational component with which the dongle 100 is engaged or from a separate power supply.

5 The memory 120 includes a dongle validation agent 128 to interact with the computational component 400 and GPS module 108 during the license validation process and various data structures, including the conversion table 132, which maps GPS coordinates against region codes, and authentication information 136, which authenticates the dongle 100 to the computational component 400. The dongle validation agent 128 provides to a
10 licensing validation agent 404 in the computational component 400 (Fig. 4) the authentication information 136, GPS timing information, and a region code (after mapping the GPS coordinates against the conversion table 132). The conversion table 132 typically is a listing of GPS coordinate ranges (e.g., a range of latitudes and a corresponding range of longitudes) against a corresponding region code. The authentication information 136
15 typically includes a unique identifier associated with the dongle 100 and/or computational component, such as a serial number, a MAC address, a secret and/or public algorithm and/or key, a digital certificate, and the like. Preferably, authentication is done using a secret, such as a secret algorithm, key, and/or certificate. Authentication is important as it prevents an unauthorized dongle from being used with the computational component. In a preferred
20 configuration, the identifier is a serial number of a board in the computational component.

Notwithstanding the specific configuration of the dongle 100 in Fig. 1, it is to be understood that the dongle can be in a multitude of other configurations. The dongle can be any hardware key that attaches or otherwise communicates with a computational component

and that must be present (or in communication with the component) to run or execute a particular piece of software and/or operate hardware. It may be programmable or non-programmable. The dongle can be a smart card, PCI board, or other electronic device.

5 The joint operation of the licensing validation and dongle validation agents 404 and 128 will now be described with reference to Figs. 2 and 3. Fig. 2 describes the operation of the licensing validation agent 404 while Fig. 3 the operation of the dongle validation agent 128.

Referring to Fig. 2, the licensing validation agent 404 first determines in decision diamond 200 whether or not the dongle 100 is present or in communication with the computational component 400. If the dongle 100 is not in communication with the component 400, the agent 404 determines in step 204 that the license is invalid. The consequences of such a determination depend on the application but can include completely disabling the software, disabling only selected features of the software, providing a warning to administration followed by complete or partial disablement of the software after a determined period, and the like. If the dongle 100 is in communication with the component 400, the agent 404 proceeds to step 208.

15 In step 208, the agent 404 requests authentication information from the dongle validation agent 128. In step 300 of Fig. 3, the dongle validation agent 128 receives the request and, in step 304, retrieves the authentication information and sends the information to the licensing validation agent 404.

20 In decision diamond 216 (Fig. 2), the licensing validation agent 404 compares the authentication information received from the dongle validation agent 128 with the authentication information stored in the memory (not shown) of the component 400. If the

differing sets of authentication information do not match, the agent 404 proceeds to step 204.

If they are identical, the agent 404 proceeds to step 217.

In step 217, the validation agent 404 requests GPS timing information from the dongle validation agent 128. In step 305 of Fig. 3, the dongle validation agent 128 receives the request and, in step 306, obtains the GPS timing information from the GPS module 108. The GPS timing information is sent to the validation agent 404 in step 307. When the GPS timing information is received by the validation agent 404 in step 218, the validation agent 404 in decision diamond 219 compares the timing information (e.g., clock setting) against the expiration date of the license. When the timing information corresponds to a time after the expiration date, the agent 404 proceeds to step 204. When the timing information corresponds to a time before the expiration date and within the license term (e.g., after the start time of the license term), the agent 404 concludes that the license is unexpired and proceeds to decision diamond 220.

In decision diamond 220, the agent 404 determines if the dongle 100 is "local" to the computational component 400. As will be appreciated, a dishonest user can set up a computational system such that the computational component 400 is running in a different region or country than the region or country where the dongle 100 is located. Although having a dongle 100 in one region to activate hardware/software in another distant region is a great inconvenience, such license abuse is not inconceivable if the cost savings to the user are large. To address this concern, the dongle 100 is configured to prohibit operation over a large network. Specifically, it can be configured to operate only as a local device without network capabilities. For example, the dongle 100 is not allowed to have an IP or MAC address that is not within a range of IP addresses or MAC addresses, respectively, defining

the computational component or a network or subnetwork containing the component or may be required to have an IP address that is the same as a network interface card (not shown) associated with the computational component 400 (but can have a port number that is different from the port number of the card). Alternatively, the communication protocol between the computational component 400 and the dongle 100 could be designed to work only in a low-latency environment, such that the protocol would fail if the physical separation (as embodied by the communication medium 408) between the component 400 and the dongle 100 (Fig. 4) were too great. For example, a maximum time delay between the time a request is made by the component 400 to the dongle 100 and the time a response to the request is received by the component 400 from the dongle 100 may be specified. When the time delay between request and response equals or exceeds the maximum time delay, the dongle 100 is not considered to be local to the computational component. Likewise when the time delay is less than the maximum time delay, the dongle 100 is considered to be local.

When the dongle 100 is not found to be local to the component 400, the agent 404 proceeds to step 204.

When the dongle 100 is found to be local to the component 400, the agent 404 proceeds to step 224. In step 224, the agent 404 requests geographic location information from the dongle validation agent 128. In step 308 of Fig. 3, the request is received by the dongle validation agent 128. In steps 312, 316, and 320 of Fig.3, the dongle validation agent 128 respectively obtains the geographic coordinates from the GPS module 108, converts the geographic coordinates into a corresponding region code by mapping the coordinates against the conversion table 132, and sends the region code to the licensing validation agent 404.

Returning again to Fig. 2, the agent 404 in decision diamond 232 determines whether

the dongle's geographic location is within a predetermined region or collection of region codes. As will be appreciated, regions having similar pricing structures can be grouped together for purposes of acceptable areas of operation for software purchased in any of the similarly priced regions. Moreover, when the computational component is purchased in a region having a higher pricing structure it can be used in different regions having a lower pricing structure but not vice versa. Thus, if a customer purchased software at a lower price in China, it could not use the software in the U.S. where the software is selling at a higher price. However, if the customer purchased the software in the U.S. it could use the software in China. If not, the agent 404 proceeds to step 204. If so, the agent 404 proceeds to step 236 and determines that the license has been successfully validated. This determination permits the computational component to operate as permitted by the terms of the license.

After either of steps 204 or 236, the agent 404 proceeds to step 240 and waits a predetermined period. After the predetermined period, the validation process described above is repeated. The duration of the predetermined period depends on the application.

A number of variations and modifications of the invention can be used. It would be possible to provide for some features of the invention without providing others.

For example, the division of the various functions performed by the validation agents 128 and 404 can be different. For example, the dongle validation agent 128 can perform authentication (steps 208-216), locality determination (step 220), and/or geographic location permissibility (decision diamond 232) and notify the licensing validation agent 404 of the result.

The steps of Figs. 2 and 3 can be performed in a different order. For example, steps 208-216 can be performed after decision diamond 220. Steps 224-232 can be performed

before either of steps 208-216 and decision diamond 220.

The licensing validation agent and dongle validation agent can be implemented as software, hardware, or a combination thereof. The dongle validation agent can, for example, be implemented as firmware.

5 In another embodiment, the GPS module is located in the computational component and a key may or may not be used. In this embodiment, the licensing validation agent interfaces directly with the GPS module and maps the GPS coordinates to a collection of region codes.

10 In yet another embodiment, the requirements for a permissible use of the computational component are not stipulated by a license. For example, the requirements can be stipulated in laws, regulations, policies or rules.

15 The present invention, in various embodiments, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various embodiments, subcombinations, and subsets thereof. Those of skill in the art will understand how to make and use the present invention after understanding the present disclosure. The present invention, in various embodiments, includes providing devices and processes in the absence of items not depicted and/or described herein or in various embodiments hereof, including in the absence of such items as may have been used in previous devices or processes, e.g., for improving performance, achieving ease and/or
20 reducing cost of implementation.

The foregoing discussion of the invention has been presented for purposes of illustration and description. The foregoing is not intended to limit the invention to the form or forms disclosed herein. Although the description of the invention has included description

of one or more embodiments and certain variations and modifications, other variations and modifications are within the scope of the invention, e.g., as may be within the skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative embodiments to the extent permitted, including
5 alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.